<center>Remarks</center>

Claims 1-7 are now pending in this application. Applicants have amended claims 1 and 7 to clarify the claimed invention. Applicants respectfully request favorable reconsideration of this application.

The Examiner rejected claims 1-7 under 35 U.S.C. § 112, first paragraph, as not being supported by an enabling specification. In particular, the Examiner asserted that the specification does not enable revalidating the compiler for any errors introduced between the first and second compilation. Applicants have amended claims 1 and 7 to recite that the first and second software are compared to find whether errors were introduced between the first and second compilations. This is clearly described in the specification, for example, in the abstract and in paragraph 0013 of the published application. Accordingly, Applicants submit that claims 1-7 comply with 35 U.S.C. § 112, first paragraph, and respectfully requests withdrawal of this rejection.

The Examiner rejected claims 1, 2, 6, and 7 under 35 U.S.C. § 103(a) as being upatentable over U.S. patent 5,754,860 to McKeeman et al. in view of U.S. published patent application 2002/0046397 to Schmitt. The Examiner rejected claims 3 and 4 under 35 U.S.C. § 103(a) as being upatentable over McKeeman et al. in view of Schmitt and further in view of U.S. published patent application 2003/0135842 to Frey et al. The Examiner rejected claim 5 under 35 U.S.C. § 103(a) as being upatentable over McKeeman et al. and Schmitt in view of Frey et al. and further in view of U.S. patent 6,598,074 to Moller et al.

<center>6</center>

The combination of McKeeman and Schmitt does not suggest the invention recited in claims 1 or 7 since, among other things, the combination does not suggest compiling a test program twice with one compiler to detect whether errors were introduced between the first and second compilings. Along these lines, McKeeman does not suggests compiling a test program a second time in a second compiler. In fact, McKeeman requires the use of "two or more compilers". Additionally, neither McKeenan nor Schmitt suggests enabling execution of a program in a device with safety features for control of real world entities if revalidation of a compiler indicates no errors in the compiler and the compiler environment. Executing a program in a controller in safety related systems is not an insignificant element of the claimed invention. Schmitt suggests a graphic flowchart debugger but does not suggest a step of enabling execution of a program after confirming that no errors have been introduced in the compiler or compiler environment.

Schmitt suggests an editor for debugging flowchart based computer programs for industrial controllers, motion controllers, but does not suggest executing a user-written program in a device with safety features for control of real world entities. Schmitt appears to suggest debugging/editing of a program using a flowchart editor may take place in a controller, as described at paragraph 0069. Schmitt only suggests that a user written program may be run in a controller for the purposes of debugging but does not suggest testing a compiler. Rather, Schmitt only suggests testing a program to find bugs. Thus, Schmitt does not relate to the same technical field as the claimed invention. Along these lines, a compiler and a user-written program are not remotely the same functional entities. Additionally, Schmitt does not suggest error testing

compilers. One skilled in the art seeking to validate a compiler would not look to a method of software debugging for solutions to problems in identifying compiler errors.

Furthermore, due to the above-described differences between McKeeman and Schmitt, it would not be obvious to combine the references. Along these lines, Schmitt does not suggest a comparison of results. Schmitt only suggests a flowchart editor for debugging a program that runs on a program controller. On the other hand, McKeeman suggests software testing using a differential testing technique to test compilers. The two are quite different.

The combination of McKeeman, Schmitt and Frey et al. does not suggest the invention recited in claims 3 and 4 since, among other things, the combination does not suggest carrying out in a device with safety features a comparison of software elements generated by two different compilings of a test program with one compiler. Along these lines, McKeeman suggests a CPU that runs two language processors 1 and 2 in a main memory 28. It is unclear whether a comparison of two output files 56a and 56b also runs in the main memory. In fact, McKeeman does not describe where the comparison is physically or functionally run. Additionally, Schmitt does not suggest performing a comparison in a device with safety features.

Furthermore, it is not obvious to combine Frey and McKeeman since, among other things, Frye relates to file tampering, where an original file has subsequently been altered in some way. Along these lines, Frey suggests a method to determine if a source code file has been tampered with, with respect to embedded system infrastructure components (ESIC). On the other hand, McKeeman relates to software testing using two compilers. Comparing outputs of

two compilers is a different technical problem. Just because a manual solution is described by Frey to solve one problem-file tampering- does not mean that use of the same solution to solve a different problem is obvious.

The combination of McKeeman et al., Schmitt, Frey et al. and Moller et al. does not suggest the present invention as recited in claim 5 since, among other things, Moller et al. does not overcome the above-described deficiencies of McKeeman et al., Schmitt, and Frey et al. Along these lines, Moller et al. does not suggest a user-written program that executes in a device with safety features for control of real world entities, wherein when the user-written program is enabled it is executed in the device. The Examiner only cites Moller et al. as suggesting downloading a variable that changes over time. Such does not suggest a user-written program that executes in a device with safety features for control of real world entities, wherein when the user-written program is enabled it is executed in the device.

In view of the above, the references relied upon in the office action, whether considered along or in combination, do not suggest patentable features of the claimed invention. Therefore, the references relied upon in the office action, whether considered alone or in combination, do not make the claimed invention obvious. Accordingly, Applicants respectfully request withdrawal of the rejections based upon the cited references.

In conclusion, Applicants respectfully request favorable reconsideration of this case and early issuance of the Notice of Allowance.

If an interview would advance the prosecution of this case, Applicants urge the Examiner to contact the undersigned at the telephone number listed below.

The undersigned authorizes the Commissioner to charge fee insufficiency and credit overpayment associated with this communication to Deposit Account No. 22-0261.

Respectfully submitted,

Date: September 4, 2009

/Eric J. Franklin/
Eric J. Franklin, Reg. No. 37,134
Attorney for Applicants
Venable LLP
575 7th Street, NW
Washington, DC 20004
Telephone: 202-344-4936